Brief Summary of the AlphaGo Nature Paper

Mastering the game of Go with deep neural networks and tree search by Silver et al. describes a new algorithm that plays a game of Go. The algorithm, called AlphaGo, is capable for the first time in history of defeating a professional human Go player. Most of today's strong Go programs are based on Monte Carlo tree search (MCTS), which predicts the opponent's move based on past observations, then uses the outcome to narrow the search to high-probability actions.

In contrast, AlphaGo uses deep convolutional neural networks (DCNN) to reduce the search tree by evaluating positions using a value network, then sampling actions using policy network. DCNN uses many layers of neurons arranged in overlapping tiles to construct abstract localized representation of an image. AlphaGo uses it by processing board positions as images of the boards and use convolutional layers to construct a representation of the position. The neural networks are trained with multiple stages of machine learning.

The first stage trains a fast rollout policy and a supervised learning (SL) policy network. SL policy network is a 13-layer network generated using supervised learning from past expert moves.

The second stage trains a reinforcement learning (RL) policy network, which uses policy gradient reinforcement learning to improve the policy network by playing against itself: the algorithm plays games between the current and a randomly selected previous iteration of the policy network.

In the third stage, a value function estimate is generated from the RL policy network. Then a value network is trained by a new self-play data set consisting of 30 million positions from separate games played between the RL policy network and itself.

Once the training is completed, the policy and value networks are combined using MCTS. The algorithm traverses the search tree consisting of action value, visit count, and prior probability. Action that maximizes the score is selected at each step of the simulation. The score consists of action value plus a bonus that is proportional to the prior probability but decays based on repeated visits. The leaf node is evaluated by combining the values from the value network and the fast rollout policy. Once the search is complete, the algorithm chooses the most visited move from the root position.

AlphaGo was evaluated by playing against other Go programs, including the strongest commercial programs Crazy Stone and Zen and the strongest open source programs Pachi and Fuego. In addition, AlphaGo played against the open source program GnuGo, which uses search methods that preceded MCTS. AlphaGo won 494 out of 495 games (99.8%) against these programs. Even when AlphaGo was given 4 handicap stones, AlphaGo won 77%, 86%, and 99% of such games against Crazy Stone, Zen and Pachi, respectively. Finally, AlphaGo played against Fan Hui, a professional 2 dan that won the European Go championships 3 times, and won all 5 matches.